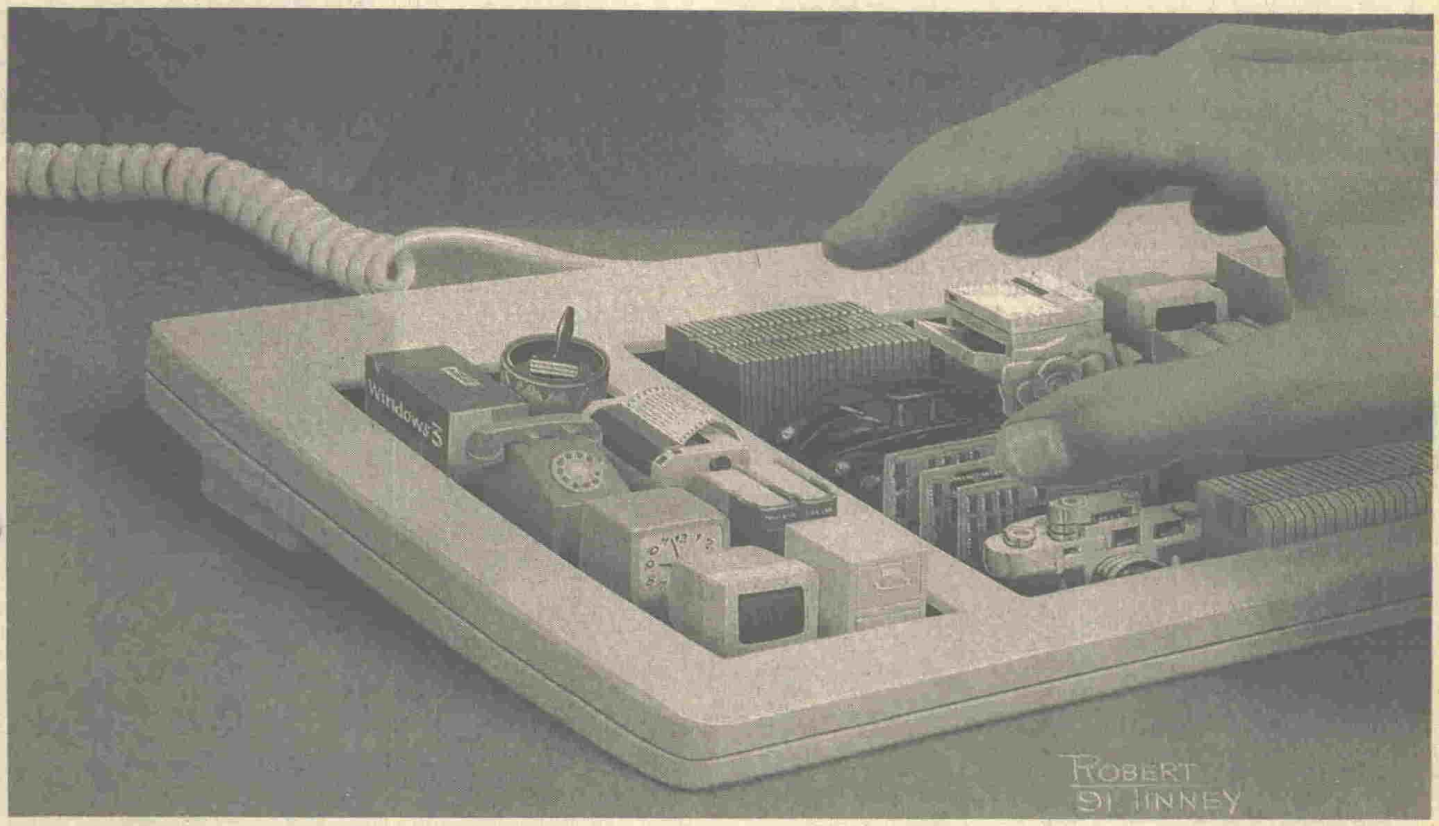
คีย์บอร์ด รหัสมอร์ส



สุดยอดแห่งการคีย์ข้อมูลที่ต้องใช้รหัส เพื่อลดความยุ่งยากในการใช้งานและอำนวยความสะดวกมากขึ้นขอแนะนำโปรแกรมสำหรับช่วยคีย์รหัสมอร์ส

การติดต่อสื่อสารในยุคปัจจุบันนั้นมีความก้าวหน้ามาก ไม่ว่าจะเป็นเรื่องความรวดเร็วและประสิทธิภาพในการรับส่งข้อมูลทำให้ผู้คนซึ่งอยู่ที่ต่าง ๆ ทุกมุมโลกได้รับทราบข่าวสารที่ทันต่อเหตุการณ์ ระบบการสื่อสารประเภทหนึ่งซึ่งปัจจุบันกำลังนิยมใช้กันอย่างแพร่หลายในเมืองไทยคือ ระบบวิทยุโทรศัพท์ ซึ่งพัฒนาไปมากไม่ว่าจะเป็นขนาดของอุปกรณ์ที่เล็กลงทำให้สะดวกในการพกพาติดตัวและประสิทธิภาพในการรับส่ง สำหรับระบบดังกล่าวเป็นระบบที่ผู้ใช้บริการสามารถติดต่อกันได้ด้วยการสนทนากันโดยตรง แต่ยังมีระบบสื่อสารอีกประเภทหนึ่งที่เคยใช้กันในอดีตซึ่งจะติดต่อกันโดยใช้รหัสสัญญาณเสียงคือ ระบบวิทยุโทรเลข ระบบสื่อสารประเภทนี้จะใช้รหัสมอร์สสำหรับติดต่อกันเป็นหลัก โดยจะต้องใช้คันเคาะเพื่อเคาะรหัสมอร์สของอักษรแต่ละตัวของข้อความที่จะส่งติดต่อกัน ซึ่งทำให้เกิดความยุ่งยากพอควร ถ้าคุณเป็นผู้หนึ่งที่ต้องใช้รหัสมอร์สเป็นประจำโปรแกรมคีย์บอร์ดรหัสมอร์สก็คงจะช่วยลดความยุ่งยากดังกล่าวได้อย่างมาก

รหัสมอร์สเป็นอย่างไร

รหัสมอร์สนั้นเป็นกลุ่มของเครื่องหมาย . (Dot) และ – (Dash) หรือเสียงสั้น (ดิ๊ด) และเสียงยาว (ด๊าด) หรืออาจใช้แสงส่งเป็นสัญญาณสั้นยาว ซึ่งกลุ่มของเครื่องหมายดังกล่าวจะนำมาเรียงกันเพื่อใช้แทนตัวอักษรต่าง ๆ ที่ใช้กันในภาษาเขียนเช่นเดียวกับรหัสแอสกีที่ใช้กลุ่มของเลขหนึ่งและศูนย์ (เลขฐานสอง) แทนตัวอักษรต่าง ๆ

จากตารางที่ 1, 2, 3 ได้แสดงรหัสมอร์สของตัวอักขระต่าง ๆ ด้วยสัญลักษณ์และตามด้วยเสียงยาวหนึ่งครั้ง โดยมีข้อกำหนดในการเคาะดังนี้ (โปรแกรมจะเป็นผู้จัดการให้)

1. เครื่องหมาย – (เสียงยาว) จะต้องยาวเป็นสามเท่าของเครื่องหมาย (เสียงสั้น)

2. ระยะเวลาที่จะต้องเว้นระหว่างเครื่องหมาย . และ – จะต้องเว้นให้ไว้นานเท่ากับความยาวของเครื่องหมาย . (เสียงสั้น)

3. ระยะเวลาที่จะต้องเว้นระหว่างตัวอักขระในข้อความที่จะส่ง จะต้องเว้นให้ไว้นานเท่ากับสามเท่าของความยาวของเครื่องหมาย . (เสียงสั้น) หรือยาวเท่ากับเครื่องหมาย –

4. ระยะเวลาที่จะต้องเว้นระหว่างคำในข้อความจะต้องเว้นให้ไว้นานเท่ากับเจ็ดเท่าของความยาวของเครื่องหมาย (เสียงสั้น)

อักษร	รหัสมอร์ส	อักษร	รหัสมอร์ส
A	.–	N	–.
B	–...	O	–––
C	–.–.	P	.––.
D	–..	Q	––.–
E	.	R	.–.
F	..–.	S	...
G	––.	T	–
H	....	U	..–
I	..	V	...–
J	.–––	W	.––
K	–.–	X	–..–
L	.–..	Y	–.––
M	––	Z	––..

ตารางที่ 2 แสดงรหัสมอร์สของตัวเลขต่าง ๆ

เลข	รหัสมอร์ส	เลข	รหัสมอร์ส
1	.————	6	–....
2	..–––	7	––...
3	...––	8	–––..
4	....–	9	––––.
5	.....	0	–––––

ตารางที่ 1 แสดงรหัสมอร์สของตัวอักษรต่าง ๆ

เครื่องหมายอื่น ๆ	รหัสมอร์ส	เครื่องหมายอื่น ๆ	รหัสมอร์ส
.	.–.–.–	_	––....–
,	––..––	/	–..–.
:	–––...	(	–.––.
?	..––..	)	–.––.–
'	.––––.	"	.–..–.

ตารางที่ 3 แสดงรหัสมอร์สของเครื่องหมายวรรคตอนต่าง ๆ

เจาะลึกตัวโปรแกรม

จากรูปที่ 1 ซึ่งแสดงตัวโปรแกรมคีย์บอร์ดรหัสมอร์สนั้น อะเรย์ CODE จะเก็บรหัสมอร์สของอักขระทั้งหมดไว้ นอกจากนี้ยังมีกลุ่มของอะเรย์ที่เก็บตำแหน่งคอลัมน์ของตัวอักษรที่แสดงเป็นโครงสร้างของคีย์บอร์ดแบบต่าง ๆ ไว้ ได้แก่ PC83POS, AT84POS, EN101POS สำหรับอะเรย์กลุ่มใหญ่ที่สุดที่เก็บรูปแบบโครงสร้างของคีย์บอร์ดซึ่งมี 3 แบบ คือ ชนิด 83 คีย์ ที่ใช้ในเครื่องรุ่น PC หรือ XT (PC83KBD), ชนิด 84 คีย์ ที่ใช้ในเครื่องรุ่น AT (AT84KBD) และชนิด 101 คีย์ (EN101KBD) ส่วนอะเรย์ KBD_DATA จะเก็บข้อมูลที่ต้องใช้ขณะที่จะแสดงโครงสร้างของคีย์บอร์ด ได้แก่ ตำแหน่งแถวและคอลัมน์เริ่มต้น, จำนวนคอลัมน์และแถวที่ต้องแสดงในแต่ละแบบ

ซอฟต์แวร์
คีย์บอร์ดรหัสมอร์ส

รูปที่ 1 แสดงตัวโปรแกรมคีย์บอร์ดรหัสมอร์ส

```
program kbd_morse;
uses
  crt, dos;
const
  code : array ['!'..'Z'] of string =
         ('.-.-.-','.-..-.','      ','...-..-','      ','      ','.----.','-.--.-','-.--.-','      ','      ','      ','--..--','-....-',
         {!     "     #     $     %     &     '     (     )     *     +     }
         '-.-.--','.-..-.','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ',
         {,     -     .     /     0     1     2     3     4     5     6     7     8     9     :     ;     <     =}
         '      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ',
         {>     ?     @     A     B     C     D     E     F     G     H     I     J     K     L     M     N     O     P     Q     R     S     T     U     V     W     X     Y     Z}
         '      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ',
         {[     \     ]     ^     _     `     a     b     c     d     e     f     g     h     i     j     k     l     m     n     o     p     q     r     s     t     u     v     w     x     y     z}
         '      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ','      ',
         {{     |     }     ~}
         '      ','      ','      ','      ');

  PC83pos : array ['!'..'Z'] of byte =
            (19,51,25,28,0,0,51,43,46,0,0,43,49,46,49,46,19,
            { !  "  #  $  %  &  '  (  )  *  +  ,  -  .  /  0  1}
             22,25,28,31,34,37,40,43,48,48,0,0,0,49,22,
            {2  3  4  5  6  7  8  9  :  ;  <  =  >  ?  @}
             21,34,28,27,26,30,33,36,41,39,42,45,40,
            {A  B  C  D  E  F  G  H  I  J  K  L  M}
             37,44,47,20,29,24,32,38,31,23,25,35,22);
            {N  O  P  Q  R  S  T  U  V  W  X  Y  Z}

  AT84pos : array ['!'..'Z'] of byte =
            (18,50,24,27,0,0,50,42,45,0,0,42,48,45,48,45,18,
             21,24,27,30,33,36,39,42,47,47,0,0,0,48,21,
             20,33,27,26,25,29,32,35,40,38,41,44,39,
             36,43,46,19,28,23,31,37,30,22,24,34,21);

  EN101pos : array ['!'..'Z'] of byte =
             (9,41,15,18,0,0,41,33,36,0,0,33,39,36,39,36,9,
              12,15,18,21,24,27,30,33,38,38,0,0,0,39,12,
              11,24,18,17,16,20,23,26,31,29,32,35,30,
              27,34,37,10,19,14,22,28,21,13,15,25,12);

  PC83kbd : array [1..567] of byte =
            (70,49,32,70,50,32,32,32,69,115,32,33,49,32,33,50,32,33,51,32,33,
             52,32,33,53,32,33,54,32,33,55,32,33,56,32,33,57,32,33,48,32,33,
             45,32,33,61,32,17,196,66,83,32,33,78,117,109,33,32,83,99,114,111,108,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             70,51,32,70,52,32,32,32,84,97,98,32,33,81,32,33,87,32,33,69,32,
             33,82,32,33,84,32,33,89,32,33,85,32,33,73,32,33,79,32,33,80,32,
             33,91,32,33,93,33,32,33,33,32,33,55,32,33,56,32,33,57,32,33,45,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,17,217,32,32,32,32,32,32,32,32,32,32,32,
             70,53,32,70,54,32,32,32,67,116,114,108,32,33,65,32,33,83,32,33,
             68,32,33,70,32,33,71,32,33,72,32,33,74,32,33,75,32,33,76,32,33,
             59,32,33,39,32,33,96,32,33,33,32,33,52,32,33,53,32,33,54,32,33,33,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,33,33,
             70,55,32,70,56,32,32,32,83,104,105,102,116,32,33,90,32,33,88,32,33,
             67,32,33,86,32,33,66,32,33,78,32,33,77,32,33,44,32,33,46,32,33,
             47,32,83,104,102,116,32,33,42,32,33,49,32,33,50,32,33,51,32,33,43,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,33,33,
             79,57,32,49,48,32,32,32,65,108,116,32,33,33,33,33,33,33,33,33,
             33,33,33,83,112,97,99,101,33,66,97,114,33,33,33,33,33,33,33,33,
             33,33,33,32,67,97,112,115,32,33,33,48,33,33,32,33,46,33,33,32,33,33)

  AT84kbd : array [1..594] of byte =
            (70,49,32,70,50,32,32,32,33,96,32,33,49,32,33,50,32,33,51,32,33,52,
             32,33,53,32,33,54,32,33,55,32,33,56,32,33,57,32,33,48,32,33,45,32,
             33,61,32,33,92,32,17,196,32,32,32,69,115,32,78,76,32,83,76,32,82,113,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             70,51,32,70,52,32,32,32,84,97,98,32,33,81,32,33,87,32,33,69,32,33,
             82,32,33,84,32,33,89,32,33,85,32,33,73,32,33,79,32,33,80,32,33,91,
             32,33,93,32,33,33,33,33,32,32,32,33,55,32,33,56,32,33,57,32,33,42,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,33,33,33,33,32,32,32,32,32,32,32,32,32,32,32,32,
             70,33,32,70,54,32,32,32,67,116,114,108,32,33,65,32,33,83,32,33,68,
             32,33,70,32,33,71,32,33,72,32,33,74,32,33,75,32,33,76,32,33,59,32,
             33,39,32,33,69,110,116,101,114,32,32,32,33,52,32,33,53,32,33,54,32,33,45,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
```

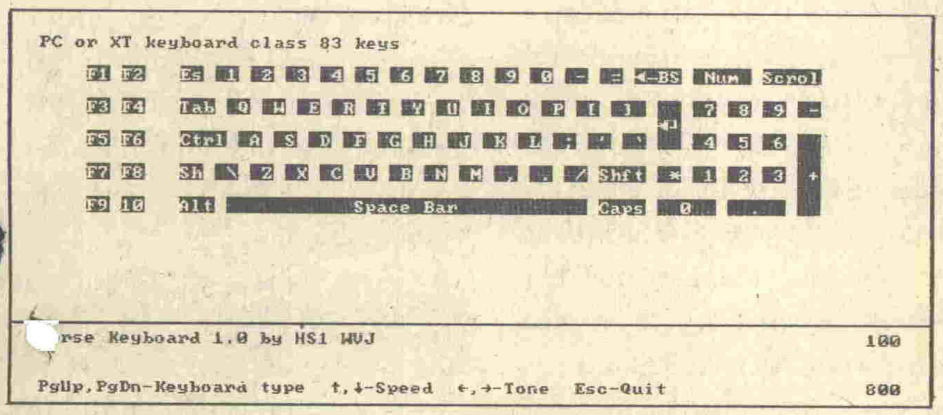
```
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,
             70,55,32,70,56,32,32,32,83,104,105,102,116,32,33,90,32,33,88,32,33,
             67,32,33,86,32,33,66,32,33,78,32,33,77,32,33,44,32,33,46,32,33,47,
             32,33,83,104,105,102,116,33,33,32,32,32,33,49,32,33,50,32,33,51,32,33,33,
             32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
```

รูปที่ 1 (ต่อ)

```
  regs : registers;
begin
  write(#218);
  for i := 1 to 78 do write(#196);
  write(#191);
  for i := 2 to 24 do
  begin
    gotoxy(1,i);  write(#179);
    gotoxy(80,i); write(#179)
  end;
  write(#192);
  for i := 1 to 78 do write(#196);
  regs.bh := 0;
  regs.cx := 1;
  regs.al := 217;
  regs.ah := 10;
  intr($10,regs);
  gotoxy(1,20);
  write(#195);
  for i := 1 to 78 do write(#196);
  write(#180);
  gotoxy(4,21); write('Morse Keyboard 1.0 by HS1 WVJ');
  gotoxy(75,21); write(dot:3);
  gotoxy(4,24); write('PgUp,PgDn-Keyboard type  ');
  write(#24#44#25,'-Speed  ',#27#44#26,'-Tone  Esc-Quit');
  gotoxy(74,24); write(notefreq:4);
  gotoxy(1,1)
end;

procedure writecolor(b : char);
begin
  if b <> ' ' then
  begin
    textcolor(black);
    textbackground(lightgray);
    if b = '!' then b := ' '
  end;
  write(b);
  textcolor(blue);
  textbackground(cyan)
end;

procedure writeblink(c : char);
const
  nocode : set of byte = [42,43,60,61,62];
begin
  if not (byte(c) in nocode) then
  begin
    textcolor(green+blink);
    textbackground(lightgray);
    write(' ',c);
    delay(80);
    textcolor(black);
    gotoxy(wherex-2,wherey);
    write(' ',c);
    textcolor(blue);
    textbackground(cyan)
  end
end;

procedure kbd_lay_out(kbd_type,x,y,cx,cy : byte);
var
  c, i, j : byte;
begin
  window(2,2,79,19);
  clrscr;
  dec(x); dec(y);
  for i := 1 to cy do
  begin
    gotoxy(x,y);
    for j := 1 to cx do
    begin
      case kbd_type of
        1 : c := PC83kbd[j+(i-1)*cx];
        2 : c := AT84kbd[j+(i-1)*cx];
        3 : c := EN101kbd[j+(i-1)*cx]
      end;
      writecolor(char(c))
    end;
    inc(y)
  end;
  gotoxy(3,2);
  case kbd_type of
    1 : write('PC or XT keyboard class 83 keys');
    2 : write('AT keyboard class 84 keys');
    3 : write('Enhanced keyboard class 101 keys')
  end;
  window(1,1,80,25)
end;

procedure getkey(var scan,ascii : byte);
var
  regs : registers;
begin
  regs.ah := 0;
  intr($16,regs);
  ascii := regs.al;
  scan := regs.ah
end;

procedure beep(c : char);
var
  time : word;
begin
  sound(notefreq);
  if c = '~' then
    time := hyphen
  else
    time := dot;
  delay(time);
  nosound;
  sound(hitfreq);
  delay(endhit);
  nosound
end;

procedure charbeep(c : char);
var
  i : byte;
  cc : char;
begin
  cc := upcase(c);
  for i := 1 to length(code[cc]) do
    beep(code[cc][i]);
  delay(endcht)
end;

procedure prepare;
begin
  clrscr;
  cursor(0);
  kbd_type := 1;
  notefreq := 800;
  hitfreq  := 32767;
  dot      := 100;
  hyphen   := 300;     { hyphen =.3dot }
  endcht   := 300;
  endhit   := 100;
  border;
  textcolor(blue);
  textbackground(cyan)
end;

procedure main_proc;
var
  row, col : byte;
begin
  kbd_lay_out(kbd_type,kbd_data[kbd_type][1],kbd_data[kbd_type][2],
                       kbd_data[kbd_type][3],kbd_data[kbd_type][4]);
  repeat
    getkey(scan,ascii);
    ascii := byte(upcase(char(ascii)));
    if ascii = 0 then
    begin
      case scan of
        67 : beep(',');  {F9 }
        68 : beep('-');  {F10}
        73 : begin { PgUp }
               dec(kbd_type);
               if kbd_type = 0 then kbd_type := 3;
               kbd_lay_out(kbd_type,kbd_data[kbd_type][1],kbd_data[kbd_type][2],
                                    kbd_data[kbd_type][3],kbd_data[kbd_type][4])
             end;
        81 : begin { PgDown }
               inc(kbd_type);
               if kbd_type = 4 then kbd_type := 1;
               kbd_lay_out(kbd_type,kbd_data[kbd_type][1],kbd_data[kbd_type][2],
                                    kbd_data[kbd_type][3],kbd_data[kbd_type][4])
             end;
        72 : begin { Up arrow }
               dec(dot,10);
               if dot <= 10 then dot := 180;
               gotoxy(75,21); write(dot:3);
               hyphen := 3*dot;
```

รูปที่ 1 (ต่อ)

```
                endcht := hyphen;
                endhit := dot
              end;
        80 : begin
                inc(dot,10);
                if dot >= 190 then dot := 10;
                gotoxy(75,21); write(dot:3);
                hyphen := 3*dot;
                endhit := dot
              end;
        75 : begin { Left arrow }
                dec(notefreq,100);
                if notefreq <= 100 then notefreq := 1200;
                gotoxy(74,24); write(notefreq:4)
              end;
        77 : begin { Right arrow }
                inc(notefreq,100);
                if notefreq >= 1300 then notefreq := 100;
                gotoxy(74,24); write(notefreq:4)
              end
      end;
    end
    else
    begin
      if char(ascii) in varid_c then
      begin
        row := kbd_data[kbd_type][2];
        if ascii in row_2 then inc(row,2);
        if ascii in row_3 then inc(row,4);
        if ascii in row_4 then inc(row,6);
        case kbd_type of
          1 : col := PC83pos[char(ascii)];
          2 : col := AT84pos[char(ascii)];
          3 : begin
                col := EN101pos[char(ascii)];
                inc(row,3)
              end
        end;
        gotoxy(col,row);
        writeblink(char(ascii));
        charbeep(char(ascii))
      end
      else if ascii = 32 then delay(7*dot)
    end
  until ascii = 27
end;

procedure clear;
begin
  clrscr;
  cursor(1)
end;

begin
  prepare;
  main_proc;
  clear
end.
```



รูปที่ 2 แสดงจอภาพหลังจากรันโปรแกรมคีย์บอร์ดรหัสมอร์ส

กลุ่มตัวแปรเซต ROW_2, ROW_3, ROW_4 นั้นใช้เก็บรหัสแอสกีของตัวอักษรที่แสดงอยู่ในแถวที่ 1, 2, 3 ในโครงสร้างของคีย์บอร์ดตามลำดับ

โพรซีเดอร์หลักของโปรแกรมเห็นจะได้แก่ MAIN_PROC ซึ่งมีหน้าที่หลักในการตรวจสอบการกดคีย์ของผู้ใช้และทำงานตามความหมายของคีย์เหล่านั้น ได้แก่ F9, F10, PgUp, PgDn, คีย์ลูกศรบน, ล่าง, ซ้าย, ขวา และตัวอักษรต่าง ๆ ที่มีรหัสมอร์ส

สำหรับโพรซีเดอร์สำคัญของโปรแกรมที่จะกล่าวถึงคือ BEEP และ CHARBEEP ซึ่งอยู่ตั้งแต่บรรทัดที่ 275-289 นั้น โพรซีเดอร์ BEEP มีหน้าที่ส่งสัญญาณเสียงสั้นหรือยาวขึ้นกับว่าตัวอักษรที่ส่งมาเป็น ``.'' หรือ ``-'' ส่วนโพรซีเดอร์ CHARBEEP จะส่งสัญญาณเสียงตามรหัสมอร์สของตัวอักษรที่ส่งมา เมื่อทำการรันโปรแกรมคีย์บอร์ดรหัสมอร์สแล้วจะแสดงดังรูปที่ 2

คุยกันก่อนจบ

โปรแกรมคีย์บอร์ดรหัสมอร์สในบทความนี้คงช่วยให้คุณที่ต้องใช้รหัสมอร์สบ่อย ๆ ใช้รหัสมอร์สได้สะดวกมากขึ้น นอกจากนี้ยังสามารถปรับความเร็วในการส่งและความถี่ของเสียงที่ส่งออกมาได้หลายระดับ หวังว่าคงถูกใจคุณโดยเฉพาะนักวิทยุสมัครเล่นขั้นกลางนะครับ